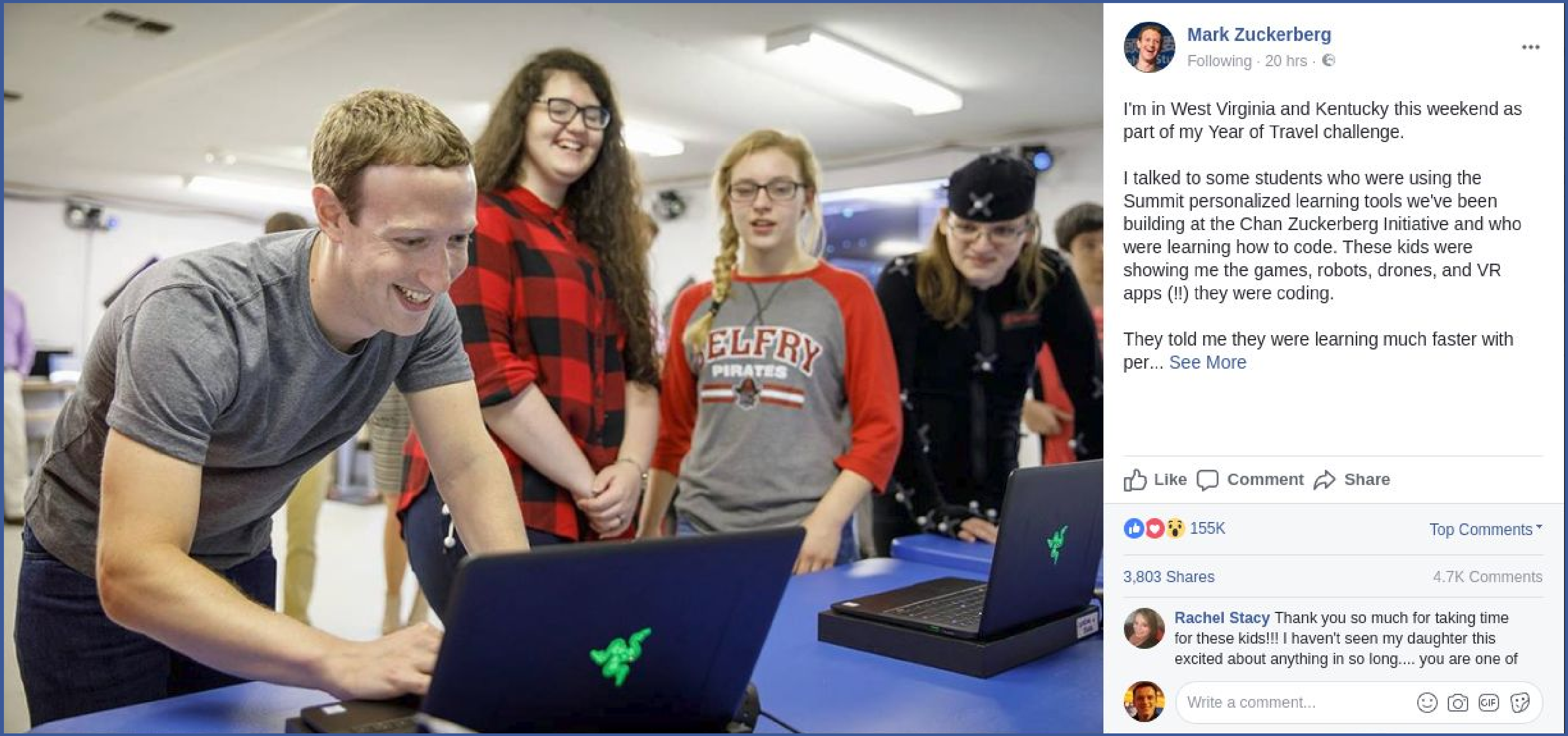
Learning-Based Probabilistic Programming Tools

Facebook Fellows Research Workshop
28th September 2017, Menlo Park, CA

Pavol Bielik
Software Reliability Lab
Department of Computer Science
ETH Zurich

facebook



Mark Zuckerberg
Following · 20 hrs ·

I'm in West Virginia and Kentucky this weekend as part of my Year of Travel challenge.

I talked to some students who were using the Summit personalized learning tools we've been building at the Chan Zuckerberg Initiative and who were learning how to code. These kids were showing me the games, robots, drones, and VR apps (!!) they were coding.

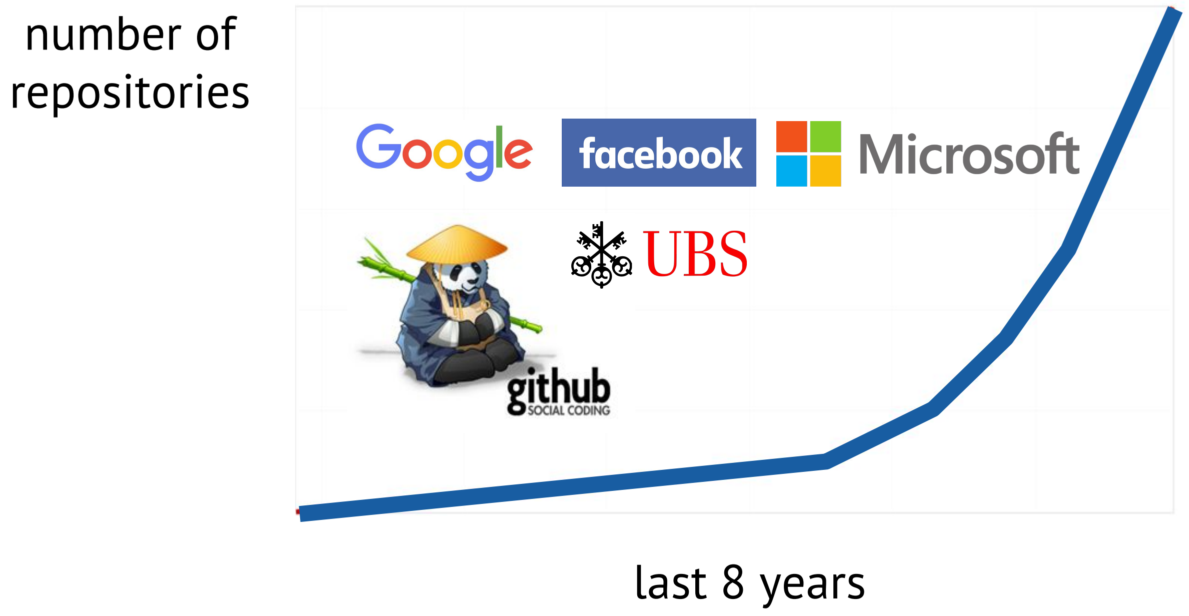
They told me they were learning much faster with per... See More

Rachel Stacy Thank you so much for taking time for these kids!!! I haven't seen my daughter this excited about anything in so long.... you are one of

```
<img class="spotlight" alt="Image may contain: 3 people, people smiling,
people sitting, laptop and indoor" src="...">
```

Vision

Create new kinds of software tools that leverage **massive codebases** to solve problems **beyond** what is possible with traditional techniques.



15 million repositories

Billions of lines of code

High quality, tested, maintained programs

Statistical Software Tools

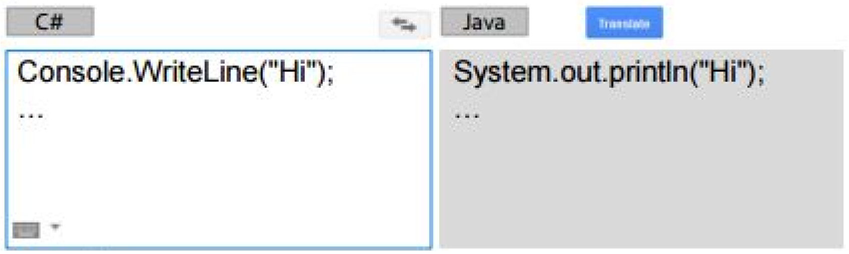
Writing Code

Code Completion

```
Camera camera = Camera.open();
camera.SetDisplayOrientation(90);
                ?
```

Porting Code

Programming Language Translation



```
Console.WriteLine("Hi");
...
```

```
System.out.println("Hi");
...
```

Program Analysis

Points-to/Type Analysis

```
function collect(val, idx, obj) {
  if (val >= this.threshold) { ... }
}

dat.filter( collect, ctx );
```

points-to

Testing/Debugging

Statistical Bug Detection

```
...
for x in range(a):
     print a[x]
```

likely error

All of these benefit from the “Big Code” and lead to applications not possible with previous techniques

Program Semantics

Program Synthesis

Program Syntax

Program Representation

Programming Languages

Program Semantics

Program Synthesis

Program Syntax

Program Representation

Programming Languages

+

Machine Learning

Probabilistic Models

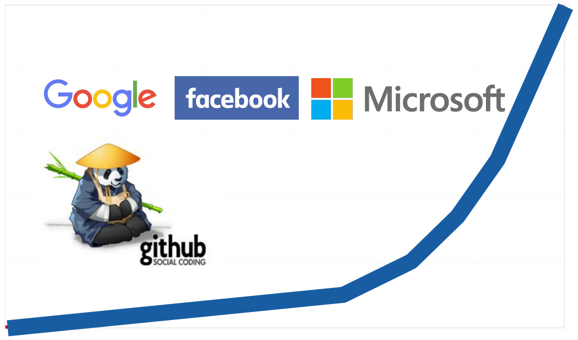
Learning/Inference Algorithms

Explainability

Precision vs Scalability

Probabilistic Model for Code

Existing Programs



Learning

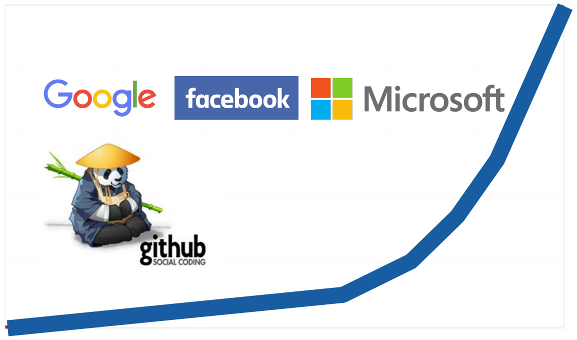
$\rightarrow$

Model

Probabilistic
Model

Probabilistic Model for Code

Existing Programs



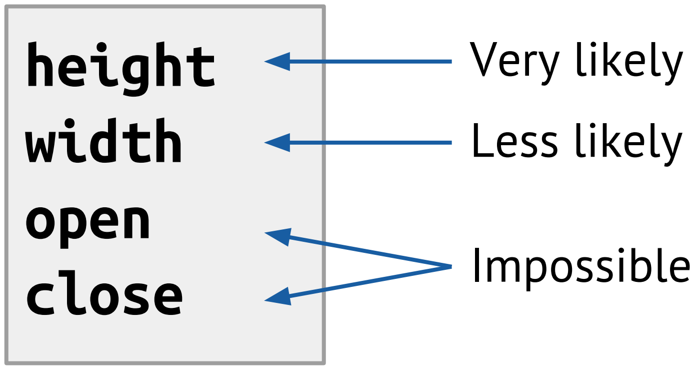
Learning

$\rightarrow$

Model

Probabilistic
Model

```
function area(a) {
  return a.width * a.
}
```



***Goal:* Assign probability to a program**

JavaScript APIs

Model	**Accuracy**
Last two tokens, Hindle et. al. [ICSE'12]	22.2%
Last two APIs, Raychev et. al. [PLDI'14]	30.4%

is this the best we can do?

JavaScript APIs

Model	Accuracy
Last two tokens, Hindle et. al. [ICSE'12]	22.2%
Last two APIs, Raychev et. al. [PLDI'14]	30.4%
Last three APIs	
Declaration Site + Last two APIs	
Variable Name + Method Name + Last API	
...	

JavaScript APIs

Identifiers
Strings
Numbers
Arguments
Properties
Statements
RegExp
Structure

JavaScript APIs

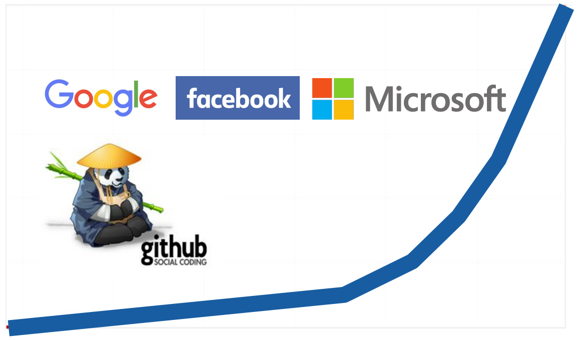
Model	**Accuracy**
Last two tokens, Hindle et. al. [ICSE'12]	22.2%
Last two APIs, Raychev et. al. [PLDI'14]	30.4%
Program synthesis	**66.4%**

Model Requirements

Existing Programs

Learning

Model



$\rightarrow$

Probabilistic Model

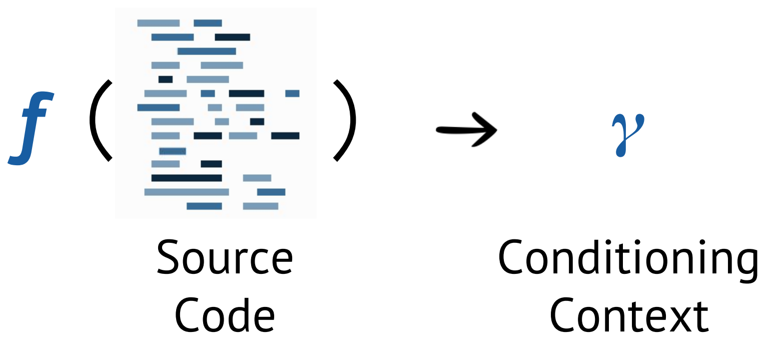
Widely Applicable

Efficient Learning

High Precision

Explainable Predictions

Program Synthesis



Synthesize a function f from a domain specific language that explains the data

Function Examples

$$f(p_1) = \{ \texttt{for}, \texttt{if}, \texttt{length==0} \}$$

```
for (j = 0; j < groups.length; j++) {
   idsInGroup = groups[j].filter(
     function(id) { return id >= 42; }
   );
  if (idsInGroup.length == 0) {
         ?
   }
}
```

$$f(p_2) = \{ \texttt{notify}, \texttt{position}, \texttt{hide} \}$$

```
elem.notify( ..., {
     position: 'top',
     hide: false,
     ?
} );
```

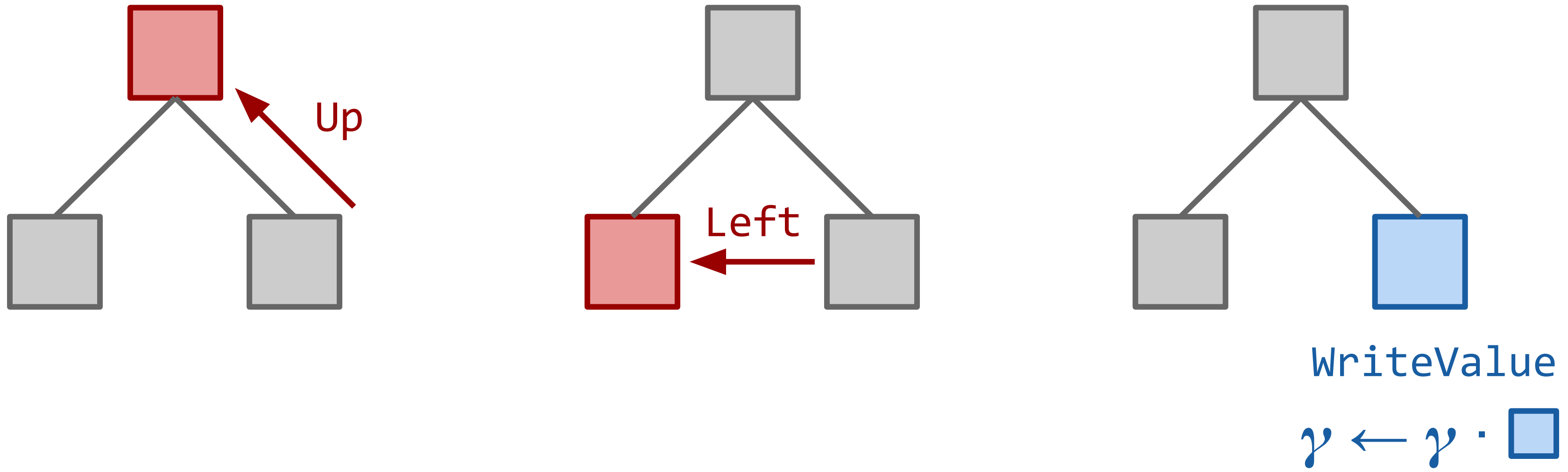
Function Representation

In general:
Unrestricted programs (Turing complete)

Our Work:
TCond Language for navigating over trees
and accumulating context

```
TCond       ::=  ε | WriteOp TCond | MoveOp TCond | BranchProg
BranchProg  ::=  if pred(x) then TCond else TCond
MoveOp      ::=  Up, Left, Right, DownFirst, DownLast,
                 NextDFS, PrevDFS, NextLeaf, PrevLeaf,
                 PrevNodeType, PrevNodeValue, PrevNodeContext
WriteOp     ::=  WriteValue, WriteType, WritePos
```

Expressing Functions: TCond Language



```
TCond       ::=  ε | WriteOp TCond | MoveOp TCond | BranchProg
BranchProg  ::=  if pred(x) then TCond else TCond
MoveOp      ::=  Up, Left, Right, DownFirst, DownLast,
                 NextDFS, PrevDFS, NextLeaf, PrevLeaf,
                 PrevNodeType, PrevNodeValue, PrevNodeContext
WriteOp     ::=  WriteValue, WriteType, WritePos
```

Example

Query TCond γ

```
elem.notify(
  ... ,
  ... ,
  {
    position: ‘top’,
    hide: false,
    $\boxed{?\quad}$
  }
);
```

Example

```
elem.notify(
   ... ,
   ... ,
   {
      position: ‘top’,
      hide: false,
      ?
   }
);
```

Query	TCond	γ
	Left	{}
	WriteValue	{hide}

Example

Query

```
elem.notify(
   ... ,
   ... ,
  {
    position: ‘top’,
    hide: false,
    ?
  }
);
```

TCond	γ
Left	{}
WriteValue	{hide}
Up	{hide}
WritePos	{hide, 3}

Example

Query

```
elem.notify(
   ... ,
   ... ,
  {
     position: ‘top’,
     hide: false,
     ?
  }
);
```

TCond	γ
Left	{}
WriteValue	{hide}
Up	{hide}
WritePos	{hide, 3}
Up	{hide, 3}
DownFirst	{hide, 3}
DownLast	{hide, 3}
WriteValue	{hide, 3, notify}

Example

Query

```
elem.notify(
   ... ,
   ... ,
  {
     position: ‘top’,
     hide: false,
     ?
  }
);
```

TCond	γ
Left	{}
WriteValue	{hide}
Up	{hide}
WritePos	{hide, 3}
Up	{hide, 3}
DownFirst	{hide, 3}
DownLast	{hide, 3}
WriteValue	{hide, 3, notify}

↓

{ Previous Property, Parameter Position, API name }

JavaScript 150k Dataset (Source Code in AST Format)

Model	**Accuracy**
Naïve Bayes	44.2%
Probabilistic Context-Free Grammars (PCFG)	51.1%
SVM	70.5%
N-gram	71.3%
Program Synthesis	**81.5%**

Linux Kernel (Source Code + Comments)

Model	Error Rate	Training	Queries/s	Size
LSTM	38.1%	~80 Hrs	300	53 MB
n-gram	35.9%	**4 Sec**	**41000**	24 MB
Synthesis	**31.4%**	8 Hrs	28000	**19 MB**

Hutter Prize Wikipedia (Natural Language + Metadata)

Model	**Bits per Character**
N-gram	1.94
Program Synthesis	**1.67**
Stacked LSTM [Graves et. al. 2013]	1.62
MRNN [Sutskever et.al. 2011]	1.60
MI-LSTM [We et.al. 2016]	1.44
HM-LSTM [Chung et. al. 2017]	1.34

Work @ ETH Zurich

Prof.
Martin
Vechev

Veselin
Raychev

Pavol
Bielik

Christine
Zeller

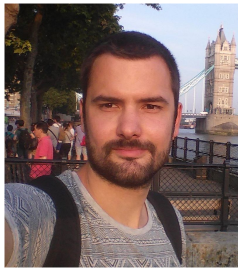
Pascal
Roos

Benjamin
Bischel

Svetoslav
Karaivanov

Benjamin
Mularczyk

Prabhakaran
Santhanam

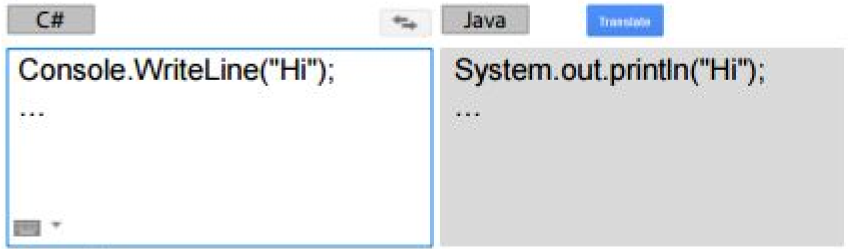

Pavle
Đorđević

Learning-Based Probabilistic Programming Tools

Writing Code
Code Completion

```
Camera camera = Camera.open();
camera.SetDisplayOrientation(90);
                ?
```

Porting Code
Programming Language Translation



```
Console.WriteLine("Hi");
...
```

```
System.out.println("Hi");
...
```

Program Analysis
Points-to/Type Analysis

```
function collect(val, idx, obj) {
  if (val >= this.threshold) { ... }
}

dat.filter( collect, ctx );
```

Points-to

Testing/Debugging
Statistical Bug Detection

```
...
for x in range(a):
    print a[x]
```

likely error

http://plml.ethz.ch/